Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 19 (2006) 122–134
Neural Networks
www.elsevier.com/locate/neunet

2006 Special issue

Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction

Vladimir M. Krasnopolsky [a,b,*], Michael S. Fox-Rabinovitz [a]

[a] Earth System Science Interdisciplinary Center, University of Maryland, College Park, MD, USA
[b] SAIC at EMC/NCEP/NOAA, 5200 Auth Road, Camp Springs MD, 20746–4304 USA

Abstract

A new practical application of neural network (NN) techniques to environmental numerical modeling has been developed. Namely, a new type of numerical model, a complex hybrid environmental model based on a synergetic combination of deterministic and machine learning model components, has been introduced. Conceptual and practical possibilities of developing hybrid models are discussed in this paper for applications to climate modeling and weather prediction. The approach presented here uses NN as a statistical or machine learning technique to develop highly accurate and fast emulations for time consuming model physics components (model physics parameterizations). The NN emulations of the most time consuming model physics components, short and long wave radiation parameterizations or full model radiation, presented in this paper are combined with the remaining deterministic components (like model dynamics) of the original complex environmental model—a general circulation model or global climate model (GCM)—to constitute a hybrid GCM (HGCM). The parallel GCM and HGCM simulations produce very similar results but HGCM is significantly faster. The speed-up of model calculations opens the opportunity for model improvement. Examples of developed HGCMs illustrate the feasibility and efficiency of the new approach for modeling complex multidimensional interdisciplinary systems.

Keywords: Neural networks; Machine learning; Numerical modeling; Climate; Weather; Hybrid model

1. Introduction

The scientific and practical significance of interdisciplinary complex environmental numerical models increased tremendously during the last decades, due to improvements in their quality via developments in numerical modeling and computing capabilities. Traditional complex environmental numerical models are deterministic models based on 'first principles' equations. For example, general circulation models a.k.a. global climate models (GCM), numerical atmospheric and oceanic models for climate and weather predictions, are based on solving time-dependent 3D geophysical fluid dynamics equations on the sphere. The governing equations of these models can be written symbolically as,

$$\frac{\partial \psi}{\partial t} + D(\psi, x) = P(\psi, x) \tag{1}$$

where ψ is a 3D prognostic or dependent variable (e.g. temperature, wind, pressure, moisture); x is a 3D independent variable (e.g. latitude, longitude, and height); D is model dynamics (the set of spectral or gridpoint 3D partial differential equations of motion, thermodynamics, continuity etc. approximated with a spectral or grid-point numerical scheme); and P is model physics (e.g. long and short wave atmospheric radiation, turbulence, convection and large scale precipitation processes, clouds, interactions with land and ocean processes, etc.) and chemistry (constituency transport, chemical reactions, etc.). These environmental models are either fully coupled atmosphere–ocean–land–chemistry–biosphere models or partially coupled models (e.g. with the chemistry component, calculated off-line, driven by the flow simulated by an atmosphere-ocean-land model).

Physical and other processes are so complicated that it is practical to include them into GCMs only as 1D (in the vertical direction) simplified or parameterized versions (usually called parameterizations). These parameterizations constitute the right-hand side forcing for the dynamics equation (1). Still, some of these parameterizations are the most time consuming components of GCMs. They are formulated using relevant first principles and observational data, and are usually based on solving deterministic equations (like radiation equations) and

* Corresponding author. Address: SAIC at EMC/NCEP/NOAA, 5200 Auth Road, Camp Springs, MD 20746-4304, USA. Tel.: +1 301 763 8000x7262; fax: +1 301 763 8545.
E-mail address: vladimir.krasnopolsky@noaa.gov (V.M. Krasnopolsky).

0893-6080/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2006.01.002

some secondary empirical components based on traditional statistical techniques like regression. Thus, for the widely used state-of-the-art GCMs all major model components are predominantly deterministic; namely, not only model dynamics but also model physics and chemistry are based on solving deterministic first principles physical or chemical equations.

Only recently attempts have been made to introduce major statistical components into GCMs, like an attempt to apply a traditional statistical technique as the expansion of hierarchical correlated functions to approximate atmospheric chemistry components (Schoendorf Rabitz & Li, 2003). This traditional statistical technique was applied successfully but with limited accuracy. Significantly higher accuracy requirements must be met for such complex multidimensional and interdisciplinary systems as modern GCMs. A particular type of machine learning technique (MLT), namely neural networks (NN), has been successfully applied for the development of new and for emulation of existing atmospheric and ocean physics parameterizations (Chevallier, Chéruy, Scott, & Chédin, 1998; Chevallier, Morcrette, Chéruy, & Scott, 2000; Krasnopolsky, Chalikov, & Rao, 2000; Krasnopolsky, Chalikov, & Tolman, 2002; Krasnopolsky, Fox-Rabinovitz & Chalikov, 2005; Krasnopolsky, Fox-Rabinovitz, & Chou, 2005; Tolman, Krasnopolsky, & Chalikov, 2005).

In this paper, we introduce hybrid numerical models which are based on a synergetic combination of deterministic numerical modeling with MLTs for emulating model physics. On the basis of the aforementioned preliminary studies and our current work with atmospheric and ocean physics, we formulate and investigate a new approach to this synergetic integration of deterministic and machine learning components in complex hybrid environmental numerical models. We discuss the conceptual and practical possibilities of developing hybrid GCM (HGCM); namely, the possibility of combining accurate and fast MLT/NN emulations of model physics components with the deterministic model dynamics of GCMs, which are the types of complex environmental models used for modern atmospheric and ocean climate modeling and weather prediction.

In Section 2 we formulate our approach to developing HGCMs using accurate and fast emulations based on MLT/NN, which allows us to significantly speed up the model calculations what is beneficial for development of high-quality high-resolution environmental numerical models. In Section 3 we present examples of HGCMs. In Section 4 we discuss the NN technique used in HGCM applications. Discussion and future plans are presented in Section 5. Section 6 contains conclusions.

2. Concept of hybrid models

One of the main problems in the development and implementation of modern high-quality high-resolution environmental models is the complexity of physical, chemical, and other processes involved. Here, we will discuss MLT emulations for model physics, keeping in mind that the approach is applicable to other model components (chemical, hydrological and other processes) as well. Parameterizations of model physics are approximate schemes, adjusted to model resolution and computer resources, based on simplified physical process equations and empirical data and relationships. Still, the parameterizations are so time-consuming, even for most powerful modern supercomputers, that some of them have to be calculated less frequently than model dynamics. Also, different physical parameterizations are calculated with different frequencies inversely proportional to their computational complexity. This may negatively affect the accuracy of environmental simulations and predictions. For example, in the case of a complex climate model like GCM, calculation of a model physics package (including the atmospheric and land physics) in a typical moderate resolution (a few degrees) GCM like the National Center for Atmospheric Research (NCAR) community atmospheric model (CAM) takes about 70% of the total model computations. This is despite the fact that while the model dynamics is calculated every 20 min, some computationally expensive parts of the model physics (e.g. short wave radiation) are calculated every hour. The most time consuming calculation of the model atmospheric physics, full long wave radiation (including calculation of optical properties), is done only every 12 h. More frequent model physics calculations, desirable for temporal consistency with model dynamics, and the introduction of more sophisticated model physics parameterizations in the future, will result in a further increase in the computational time spent for calculating model physics.

This situation is an important motivation for looking for alternative, faster (and most importantly) very accurate ways of calculating model physics, chemistry, hydrology and other processes. During the last decade, a new machine learning approach based on NN approximations or emulations was applied for accurate and fast calculation of atmospheric radiative processes (e.g. Chevallier, Chéruy, Scott, & Chédin, 1998; Krasnopolsky, Breaker, & Gemmill, 1997), and for environmental satellite data processing (Krasnopolsky, Breaker, & Gemmill, 1995; Krasnopolsky, Gemmill, & Breaker, 1999; Krasnopolsky & Schiller 2003). Recently, the NN approach has also used by the authors for emulations of model physics in ocean and atmospheric numerical models (Krasnopolsky et al., 2000, 2002, 2005) in which an acceleration in the calculation of model physics components of 10–10^5 times have been achieved as compared to the time needed for calculating the corresponding original parameterizations of model physics.

Based on the above results we introduce a new concept or design of a hybrid complex environmental model. We use NCAR CAM (see J. Climate, 1998 for the description of the model), a state-of-the-art widely recognized GCM used by a large modeling community for climate predictions, as an example of a complex environmental model. We start from the definitions of terms used for formulating our approach.

A NN *emulation* of a model physics parameterization is a functional imitation of this parameterization so that the results of model calculations with the original parameterization and with its NN emulation are physically (and climatologically) practically identical. The high quality of NN emulations

achieved is due to the high accuracy of approximation of the original parameterization.

Due to the capability of modern MLTs in providing a very high accuracy of approximating complex systems like model physics, our NN emulations of model physics parameterizations are practically identical to the original physical parameterizations. It allows us to preserve the integrity and level of complexity of the state-of-the-art parameterizations of model physics. As a result, a HGCM using these NN emulations produces climate simulations that are practically identical to those of the original GCM. This is achieved by using data for NN training that are simulated by running an original GCM, i.e. with the original parameterization. Using model-simulated data for NN training allows us to achieve a high accuracy of approximation because simulated data are free of the problems typical for empirical data (problems like high level of observational noise, a sparse spatial and temporal coverage, a poor representation of extreme events, etc.). In the context of our approach, the accuracy and improved computational performance of NN emulations and eventually the HGCM, is always measured against the original GCM and its original parameterization. It is noteworthy that the developed NN emulation has the same inputs and outputs as the original parameterization and is used precisely as its functional substitute within the model.

NN emulations of model physics are based on the fact that any parameterization of model physics can be considered as a continuous or almost continuous (i.e. with a finite number of finite discontinuities like step functions) mapping (output vector vs. input vector dependence), and NNs (multilayer perceptrons (MLP) in our case) are a generic tool to approximate such mappings (Cybenko, 1989; Funahashi, 1989; Hornik, 1991). In this study, we used a MLP with one hidden layer and with the hyperbolic tangent activation function and a linear output layer presented by Eq. (2)

$$y_q = a_{q0} + \sum_{j=1}^{k} a_{qj} \tanh\left(b_{j0} + \sum_{i=1}^{n} b_{ji} \bullet x_i\right); \quad q = 1, 2, \ldots, m \tag{2}$$

where x_i and y_q are components of the input and output vectors, respectively; a and b are weights and biases, n and m are the numbers of inputs and outputs, respectively; and k is the number of neurons in the hidden layer.

Let us formulate a developmental framework and test criteria (that according to our experience) can be recommended when developing and testing machine learning components of HGCM, i.e. NN emulations of model physics components. The developmental process consists of the *three major steps*:

1. Problem analysis or analysis of the model component (for example, the original parameterization) to be approximated to determine the optimal structure and configuration of NN emulations.
2. Generation of representative data sets for training, validation, and testing. When creating a representative data set, the original GCM is run long enough to produce all possible atmospheric model simulated states, phenomena, etc.
3. NN training: Several different versions of NNs with different architectures, initialization, and training algorithms should be trained and validated.

Testing of the HGCM using the trained NN emulation consists of two major steps. The *first step* is testing the accuracy of the NN approximation against the original parameterization using the independent test data set. Both the original parameterization and its NN emulation are complicated multidimensional objects (mappings). Many different statistical metrics of approximation accuracy should be calculated to assure that a sufficiently complete evaluation of the approximation accuracy is obtained. For example, total, level, and profile statistics have to be evaluated (see Section 3). The *second* test *step* consists of a comprehensive analysis of parallel HGCM and GCM runs. For the parallel model simulations, all relevant model prognostic (i.e. time-dependent model variables) and diagnostic fields should be analyzed and carefully compared to assure that the integrity of the original GCM and its parameterization, with all its details and characteristic features, is precisely preserved when using a HGCM with NN emulation (see Section 3). This test step involving model simulations is crucially important. GCMs are essentially nonlinear complex systems; in such systems, small systematic and even random approximation errors can accumulate over time and produce a significant impact on the quality of the model results. Therefore, the development and application framework of the new hybrid approach should be focused on obtaining a high accuracy in both NN emulation and HGCM simulations.

3. Examples of machine learning components and HGCMS

The NCAR CAM and NASA NSIPP (Natural Seasonal-to-Interannual Predictability Program) GCM are used in this study as examples of GCMs. The NCAR CAM is a spectral model which has 42 spectral components (or approximately 3–3.5° horizontal resolution) and 26 vertical levels. The NSIPP model is a grid point GCM which has $2\times2.5°$ horizontal resolution (latitude$\times$longitude) and 40 vertical levels. Note that the model vertical levels are distributed between the surface and upper stratosphere which is at approximately 60–80 km. Development of NN emulations was done for the two most time consuming components of model physics, long wave radiation (LWR) and short wave radiation (SWR). The NCAR and NSIPP models have different LWR and SWR parameterizations. The complete description of NCAR CAM atmospheric LWR is presented by Collins (2001) and Collins, Hackney, and Edwards, (2002), and of the NSIPP LWR by Chou, Suarez, Liang, and Yan (2001). The full model radiation (or total LWR and SWR) calculations take $\sim$70% of the total model physics calculations.

3.1. NCAR CAM long wave radiation

The function of the LWR parameterization in atmospheric GCMs is to calculate heating fluxes and rates produced by LWR processes. Here we give a very general and schematic outline of this parameterization in order to illustrate the complexity that makes it a computational 'bottleneck' in the NCAR CAM physics.

The method for calculating LWR in the NCAR CAM is based on LW radiative transfer equations in an absorptivity/emissivity formulation (see Collins, 2001 and references therein)

$$F^{\downarrow}(p) = B(p_t)\varepsilon(p_t,p) + \int_{p_t}^{p} \alpha(p,p')\mathrm{d}B(p')\, F^{\uparrow}(p)$$

$$= B(p_s) - \int_{p}^{p_s} \alpha(p,p')\mathrm{d}B(p') \tag{3}$$

where $F^{\uparrow}(p)$ and $F^{\downarrow}(p)$ are the upward and the downward heat fluxes; $B(p)=\sigma T^4(p)$ is the Stefan–Boltzmann relation; pressures p_s and p_t refer to the top and surface atmospheric pressures; and α and ε are the atmospheric absorptivity and emissivity. To solve the integral Eq. (3), the absorptivity and emissivity have to be calculated by solving the following integro-differential equations

$$\alpha(p,p') = \frac{\int_0^{\infty}\{\mathrm{d}B_\nu(p')/\mathrm{d}T(p')\}(1-\tau_\nu(p,p'))\mathrm{d}\nu}{\mathrm{d}B(p)/\mathrm{d}T(p)}\varepsilon(p_t,p)$$

$$= \frac{\int_0^{\infty} B_\nu(p_t)(1-\tau_\nu(p_t,p))\mathrm{d}\nu}{B(p_t)} \tag{4}$$

where the integration is over wave number ν, and $B(p_t)$ is the Planck function. To solve Eq. (4) for the absorptivity and emissivity, additional calculations have to be performed and the atmospheric transmission, τ_ν has to be calculated. This calculation involves a time consuming integration over the entire spectral range of gas absorption.

The input vectors for the NCAR CAM LWR parameterization include ten vertical profiles (atmospheric temperature, humidity, ozone, CO_2, N_2O, CH_4, two CFC mixing ratios (the annual mean atmospheric mole fractions for halocarbons), pressure, and cloudiness) and one relevant surface characteristic (upward LWR flux at the surface). The CAM LWR parameterization output vectors consist of the vertical profile of heating rates (HRs) and several radiation fluxes, including the outgoing LWR flux from the top layer of the model atmosphere (the outgoing LWR or OLR).

The NN emulation of the NCAR CAM LWR parameterization has the same number of inputs (a total of 220) and outputs (a total of 33) as the original NCAR CAM LWR parameterization. We have developed several NNs which all have one hidden layer with 20, 90, 150, 200, 250, or 300 neurons. Varying the number of hidden neurons allows us to demonstrate the dependence that the accuracy of approximation has on this parameter as well as its convergence (Krasnopolsky et al., 2005), and as a result provide an accuracy of approximation sufficient for the climate model.

NCAR CAM was run for 2 years to generate representative data sets. The first year of the model simulation was divided into two independent parts, each containing input/output vector combinations. The first part was used for training and the second one for validation (control of overfitting, control of a NN architecture, etc.). The second year of simulation was used to create a test data set completely independent from both training and validation sets. This data set was used for testing only. All approximation statistics presented in this section are calculated using this independent test data set.

Our NN emulations were tested against the original NCAR CAM LWR parameterization. Mean difference B (a bias or a systematic error of approximation) and the root mean square difference RMSE (a root mean square error of approximation) between the original parameterization and its NN emulation are calculated as follows:

$$B = \frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}[Y(i,j) - Y_{\mathrm{NN}}(i,j)]$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{L}[Y(i,j) - Y_{\mathrm{NN}}(i,j)]^2}{NL}} \tag{5}$$

where $Y(i,j)$ and $Y_{\mathrm{NN}}(i,j)$ are outputs of the original parameterization and its NN emulation correspondingly; $i=$(lat, lon), $i=1,\ldots, N$ is the horizontal location of a vertical profile; N is the number of horizontal grid points; and $j=1,\ldots, L$ is the vertical index where L is the number of the vertical levels.

All our NN emulations have almost zero or negligible systematic errors (biases), which almost do not depend on height and are indistinguishable from each other for the scale of Fig. 1. The rest of Fig. 1 shows the vertical profiles of RMSE (5) for the six developed NNs. For all NNs with the number of hidden neurons starting at 90, RMSE (which is a purely random error for the case of a zero bias) for the 10 upper levels does not exceed 0.2 K/day and reaches just about 0.6–0.8 K/day at the lowest level, which does not lead to significant errors in HGCM simulations (see below).

Because NN20 is significantly less accurate and NN250 and NN300 are not significantly more accurate than NN200, only three NNs, NN90, NN150, and NN200 are included into Table 1. Table 1 shows bulk test statistics for the approximation accuracy and computational performance for the three best (in terms of accuracy and performance) developed NN emulations. Mean values and standard deviations (σ_{HR}) of HRs are presented in the title of Table 1 for a better understanding of relative errors.

In addition to having a high approximation accuracy, our NN emulations perform about 80–35 times faster (for NN90, NN150, and NN200, correspondingly) than the original NCAR

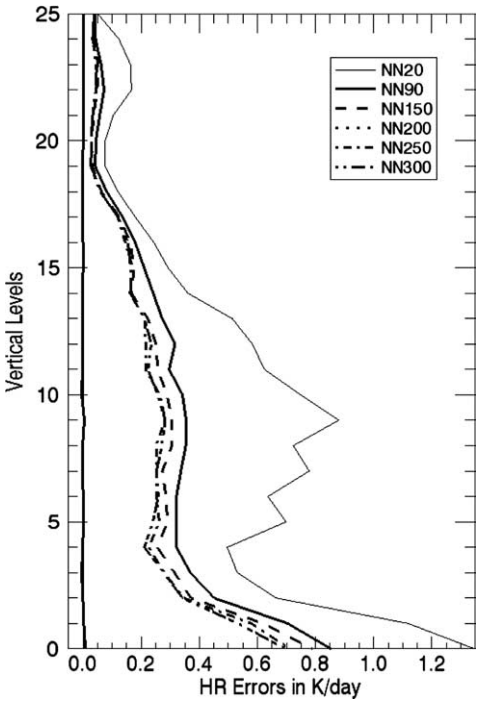

Fig. 1. LWR NN emulation errors for NCAR CAM. Vertical profiles of mean approximation errors at each of 26 model levels, the level biases (the left vertical solid line) and level RMSEs (5), for six developed NNs (NN20—thin solid, NN90—thick solid, NN150—dashed, and NN200—dotted, NN250—dash-dotted, NN300—dash-double dotted lines), all in K/day.

LWR parameterization. Table 1 and Fig. 1 clearly demonstrate a systematic improvement in approximation accuracy when increasing the size of the NN hidden layer as well as the accuracy conversion. Both the original parameterization and its NN emulation are complicated multidimensional objects (mappings). In this case, calculating bulk statistics is not sufficient for evaluating the approximation accuracy. We evaluated many different statistical metrics of approximation accuracy (Krasnopolsky et al., 2002, 2005).

The analysis of approximation errors presented above shows that the NN technique is capable of providing NN emulations with almost no systematic errors or biases and only small random errors. Obviously, the final decision on choosing the optimal version of the NN to be implemented into the model should be made based on testing these NNs in HGCM simulations (see Krasnopolsky et al., 2005). For assessing the impact of using NN emulation of the LWR parameterization in the HGCM, parallel climate simulation runs were performed with the original GCM (NCAR CAM including the original LWR parameterization) as the control run, and with the HGCM (NCAR CAM including our NN emulations of LWR described above). The climate simulations were run for 10 years (a period long enough for testing the model performance) starting after a 2 year training and validation period, namely for years 3–12.

Comparisons between the control and NN emulation runs presented in Table 2 are done by analyzing the time (10-year) and global mean differences between the results of the parallel runs, as is routinely done in climate modeling. In the climate simulations performed with the original GCM and with HGCM, the time and global mean mass or mean surface pressure is precisely preserved, which is the most important preservation property for climate simulations. For example, for the NN150 run there is a negligible difference of 0.0001% between the NN and control runs (see Table 2). Other time global means, some of which are also presented in Table 2, show a profound similarity between the simulations for these terms, with differences usually within about 0.03%. These very small differences indicate the almost identical or very close results for the parallel climate simulations. Other simulations (with NN90 and NN200) also show that the HGCM results are profoundly similar to those of the original GCM (Krasnopolsky et al., 2005).

3.2. NASA NSIPP long wave radiation

The robustness of our approach and the results obtained were investigated using another GCM. The NASA NSIPP

Table 1
Statistics estimating the accuracy of HRs (in K/day) calculations and LWR computational performance for NCAR CAM and NASA NSIPP model, both using NN emulations (in these HGCMs) vs the original LWR parameterizations (in GCMs) The total mean value for HRs = −1.36 K/day and the standard deviation $\sigma_{HR} = 1.93$ K/day. The complexity column shows the total number of weights in emulating NN

Model	Bias (K/day)	RMSE (K/day)	Complexity	Performance
NCAR NN90	-4×10^{-4}	0.33	22,893	∼80 times faster
NSIPP NN90	7×10^{-4}	0.58	22,001	
NCAR NN150	1×10^{-4}	0.28	38,133	∼50 times faster
NSIPP NN150	5×10^{-4}	0.45	36,641	
NCAR NN200	5×10^{-5}	0.26	50,833	∼35 times faster
NSIPP NN200	3×10^{-4}	0.38	48,841	

Table 2
Time (10-year) and global means for mass (mean sea level pressure) and other model diagnostics for the NCAR CAM-2 climate simulations with the original LWR parameterization (in GCM), and its NN emulation (in HGCM) using NN150 and their differences (in %)

Field	GCM with the original LWR parameterization	HGCM with NN emulation	Difference (%)
Mean sea level pressure (hPa)	1011.480	1011.481	0.0001
Surface temperature (K)	289.003	289.001	0.0007
Total precipitation (mm/day)	2.275	2.273	0.09
Total cloudiness (fractions, %)	60.7	60.9	0.03
Wind at 12 km (m/s)	16.21	16.27	0.006

GCM (with a different LWR parameterization and other model components as compared to those of the NCAR CAM and its LWR parameterization) was used for this purpose. The input vector for the NSIPP LWR includes five vertical profiles (cloud fraction, pressure, temperature, specific humidity, and ozone mixing rate) and one surface temperature, for a total of 202 inputs. The NSIPP LWR output vector consists of a profile of heating rates and one surface parameter, or a total of 41 outputs.

Approximation statistics for three NNs, namely NN90, NN150, and NN200, are included into Table 1. Table 1, Figs. 1 and 2, produced for two different LWR parameterizations used in two different GCMs, clearly demonstrate a systematic and similar improvement of the NN approximation accuracy with increasing size of the NN hidden layer. For more details about NASA NSIPP LWR NN emulations, see Krasnopolsky, Fox-Rabinovitz, and Chou (2005).

3.3. NCAR CAM short wave radiation

The second component of atmospheric radiation is the short wave radiation (SWR). LWR and SWR together comprise full radiation. The function of the SWR parameterization in atmospheric GCMs is to calculate heating fluxes and rates produced by SWR processes. The description of the NCAR CAM atmospheric SWR parameterization is presented in a special issue of Journal of Climate (1998). The input vectors for the NCAR CAM SWR parameterization include twenty one vertical profiles (specific humidity, ozone concentration, pressure, cloudiness, aerosol mass mixing ratios, etc.) and several relevant surface characteristics. We developed NN emulations for CAM-2 and CAM-3 versions of NCAR CAM SWR parameterizations. The major difference between the CAM-2 and CAM-3 SWR versions is that CAM-3 uses significantly more information about aerosols. This extended aerosol information is responsible for the substantial increase in the number of inputs into the CAM-3 SWR parameterization as compared with CAM-2. The CAM SWR parameterization output vectors consist of a vertical profile of heating rates (HRs) and several radiation fluxes.

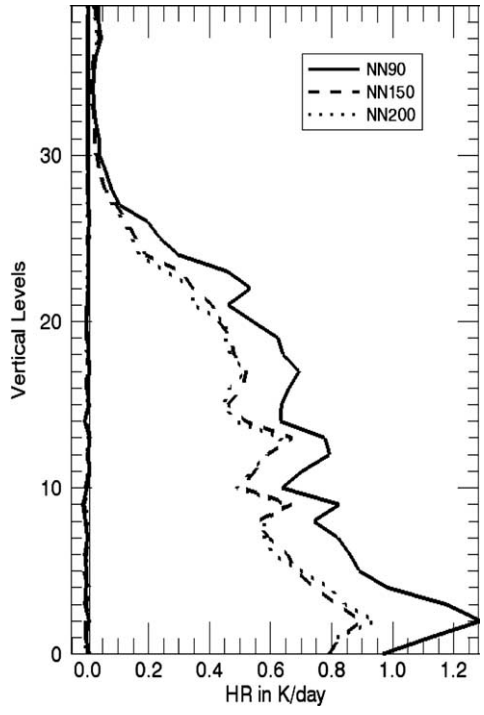

Fig. 2. LWR NN emulation errors for NASA NSIPP GCM. The vertical profiles of mean approximation errors at each of 40 model levels, for level biases (the left vertical line) and level RMSEs (5), for three developed NNs (NN90—thin solid, NN150—dashed, and NN200—dotted), all in K/day.

The NN emulations of NCAR CAM-2 and CAM-3 SWR parameterizations have 173 and 451 inputs, correspondingly, and 33 outputs, which are the same as for the original NCAR CAM-2 and CAM-3 SWR parameterizations. We have developed several NNs, all of which have one hidden layer with 50, 100, 150, or 200 neurons ($k=50$, 100, 150, and 200 in Eq. (2)).

The data sets for training, validating, and testing NNs emulating SWR were generated in the same way as those for the LWR NN emulations described above. Our SWR NN emulations were tested against the original NCAR CAM-2 and CAM-3 SWR parameterizations. Table 3 shows bulk test statistics for the accuracy of approximation and computational performance for the three best (in terms of accuracy and performance) SWR NN emulations. Mean values and standard deviations (σ_{HR}) of HRs are presented in the title of Table 3 for a better understanding of relative errors.

The vertical distributions of approximation errors for CAM-2 SWR NNs presented in Table 3 are shown in Fig. 3. The table and figure demonstrate that SWR and LWR NN emulations are similar in terms of their approximation accuracy. Parallel runs of NCAR CAM support this conclusion and show results similar to those presented in the previous section for parallel runs with the LWR NN emulation. Comparisons of the parallel runs are summarized in Table 4. See the discussion of Table 2 for comparison.

Let us now give an example of a key simulated prognostic field, temperature, and the differences (produced in the 10-year parallel runs) between the control GCM simulation and the

Table 3
Statistics estimating the accuracy of HR (in K/day) calculations and computational performance for NCAR CAM-2 and CAM-3 SWR using NN emulation (in these HGCM). using NN100, NN150, and NN200 vs. the original parameterization (in this GCM). For SWR, the mean value for HRs = 1.47 K/day and $\sigma_{HR}=1.98$ K/day. The complexity column shows the total number of weights in emulating NN

NN	Parameterization	Bias	RMSE	Complexity	Performance
NN100	SWR CAM-2	3×10^{-3}	0.17	20,733	
	SWR CAM-3	3×10^{-3}	0.18	48,533	
NN150	SWR CAM-2	3×10^{-4}	0.15	31,083	~ 8[a] times faster
	SWR CAM-3	4×10^{-4}	0.18	72,783	
NN200	SWR CAM-2	1×10^{-3}	0.14	41,433	
	SWR CAM-3	2×10^{-4}	0.15	97,033	

[a] The main reason for the smaller performance gain for SWR vs. LWR (see Table 2) when using NN emulation in the case of the SWR parameterization is that the original SWR CAM-2 parameterization is simpler and about 10 times faster than the original LWR CAM-2 parameterization.

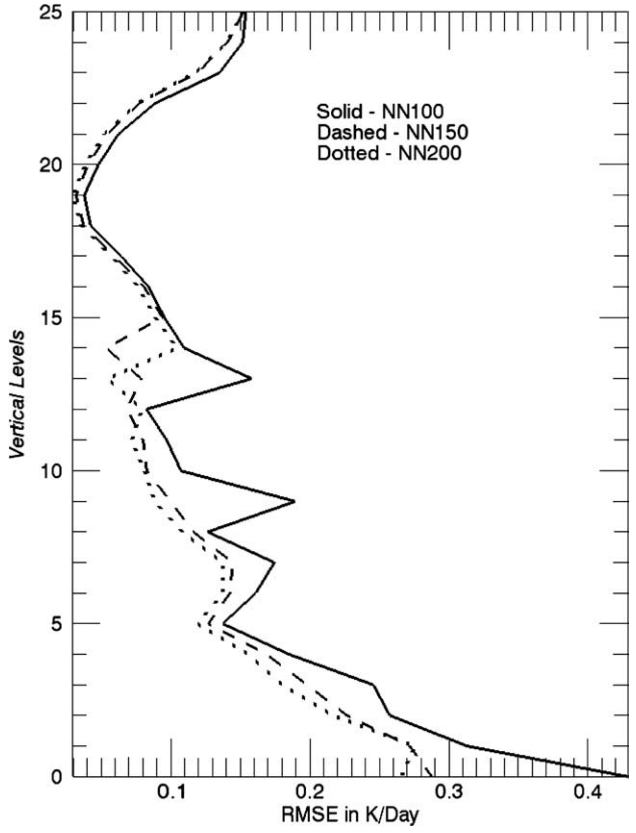

Fig. 3. SWR NN emulation errors for NCAR CAM. Vertical profiles of level RMSEs at each of 26 model levels for SWR NN emulations for three developed NNs (NN100—solid, NN150—dashed, and NN200—dotted line) in K/day.

HGCM simulation with SWR NN150 emulation. The time (10-year) mean vertical distributions of latitudinal means for temperature (T) are presented in Fig. 4. The upper left panel of Fig. 4 shows the HGCM simulation (with SWR NN150), the upper right panel the control simulation (with the original SWR parameterization), and the bottom panel shows their difference or bias. Therefore, bias is calculated against the control run. The temperature distributions for the control and NN emulation runs (see upper panels) are practically indistinguishable from each other. Such a profound pattern similarity is further confirmed and quantified by a very small, practically negligible bias shown in the bottom panel. Temperature bias for the NN90 run is mostly limited in magnitude to 0.1–0.2 K. It increases to a maximum of 0.8 K for the southern polar domain above the 200 hPa level. Just as a reference, the temperature observation errors are about 2–3 K. Therefore, bias for the HGCM run (with SWR NN150) is just a small faction of the observation error.

Table 4
Time (10-year) and global means for model diagnostics from NCAR CAM-2 climate simulations with the original SWR (in GCM), its NN emulation (in HGCM) using NN150, and their differences in %

Field	GCM with the original SWR parameterization	HGCM with SWR NN emulation	Difference (%)
Mean sea level pressure (hPa)	1011.481	1011.483	0.0002
Surface temperature (K)	289.005	288.973	0.003
Total precipitation (mm/day)	2.86	2.87	0.0009
Total cloudiness (fractions, %)	60.73	60.81	0.008
Wind at 12 km (m/s)	16.21	16.18	0.003

A detailed comparison of diagnostic and prognostic fields for the parallel runs of GCM (the control run) and HGCM (using NN emulation for the LWR parameterization) is presented in Krasnopolsky at al. (2005). They show that the parallel run fields are as close to each other as those shown in Fig. 4. Therefore, both components of radiation, LWR and SWR, can be successfully emulated using the NN approach. It means that these most time consuming components of model physics can be significantly sped up without any negative impact on the accuracy of climate simulations.

4. Specific NN aspects of the HGCM application

Here, we consider the specific aspects of the NN approach relevant to our application. The very large size and other specific features of our model component emulating NNs have determined the selection of the NN type, MLP (see the discussion in Section 2), and many other specific NN features used in our application. Some of these features are briefly discussed below.

4.1. Architecture of NN emulations: a single NN vs multiple NNs

In our approach described in Section 2, we treat an entire parameterization as an elementary/single object and emulate its functionality (input–output relationship) as a whole. Practical implementation of this approach allows for multiple solutions in terms of the number of NNs used for emulation. The MLP NN presented by Eq. (2) can be implemented as a single NN with m outputs, m NNs with one output each, or several NNs with the total number of outputs equal to m.

A single emulating NN per parameterization solution is convenient because of the simplicity of its design. It also has a great advantage in terms of speeding up the calculations when, as in our application, the outputs of the parameterization and, therefore, the outputs of the emulating NN are highly correlated. In the case of a single NN (2) with many outputs, all outputs are built from the same hidden neurons; they are different linear combinations of the same neurons. Fewer neurons are required to approximate a particular number of correlated outputs than to approximate the same number of uncorrelated ones. Thus, in the case of correlated outputs, one NN per emulation solution provides a significantly higher performance at the same approximation accuracy than a battery of m NNs, each with one output.

On the other hand, a single emulating NN per parameterization solution is more complicated in terms of NN training. It leads to a higher dimensionality of the training space. For example, for CAM-3 SWR NN, the dimensionality of the training space (the total number of NN weights) exceeds

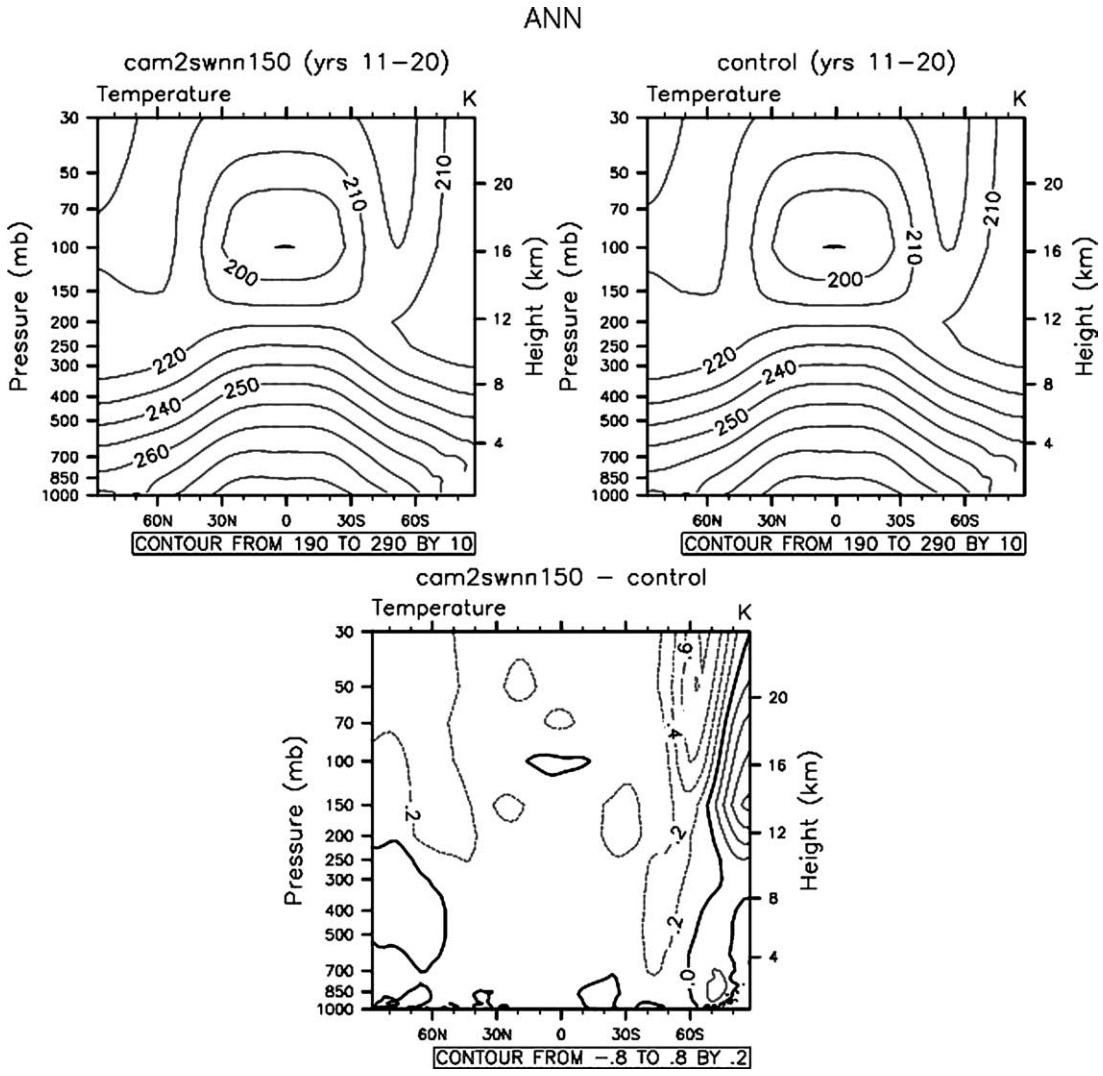

Fig. 4. NCAR CAM time (10-year) and latitudinal mean vertical distributions of temperature (in K) for HGCM with SWR NN150, control (GCM with the original SWR) runs (the upper left and right panels, respectively), and their difference (the bottom panel). The contour intervals are 10 and 0.2 K for the upper and bottom panels, respectively.

100,000 if the number of hidden neurons exceeds 200. Training such a complex NN is very slow and reaching a good local minimum is not guaranteed. As a result, the improvement in approximation accuracy with an increase of the number of hidden neurons may be slow and not monotonic (see Krasnopolsky et al., 2005; Fig. 1).

A battery of m single-output NNs is slower than a single NN emulation with m outputs; however, because each of m NNs is simpler than a single NN emulation with m outputs, training these multiple NNs may be a simpler and faster procedure than that for a single NN emulation with m outputs. Higher approximation accuracy may be obtained more easily with multiple NNs if performance can be traded off. Fig. 5 shows a comparison of absolute and relative approximation errors for three single LWR emulating NNs (2), each with 150 neurons in one hidden layer and 33 outputs (solid, dashed, and dotted lines), with a battery of 33 NNs with a single output each (dash-dotted line). The difference between the three single NNs is due to the different normalization of outputs (see Section 4.2). The accuracy of the battery of 33 NNs is significantly higher than that of the single NN; however, the total number of hidden neurons in the NN battery is about 950. As a result, the NN battery performs about six times slower than a single NN with 150 hidden neurons and only about eight times faster than the original LWR parameterization. The training space for a single NN with 950 hidden neurons would have a dimensionality of about 250,000 (only about 38,000 for a NN with 150 neurons). A battery of 33 NNs with an approximation accuracy close to that of a single NN with 150 neurons (e.g. Fig. 5, solid line) has a total of about 400 hidden neurons; therefore, the correlation of outputs in the case of the LWR parameterization allows us to obtain a performance gain of about 2.5–3 times (for the same approximation accuracy) when using a single NN with multiple outputs.

The possible choices among many topological solutions, from a single NN with m outputs to m single-output NNs demonstrate an important flexibility in the NN emulation technique that offers a speed vs accuracy trade-off. This

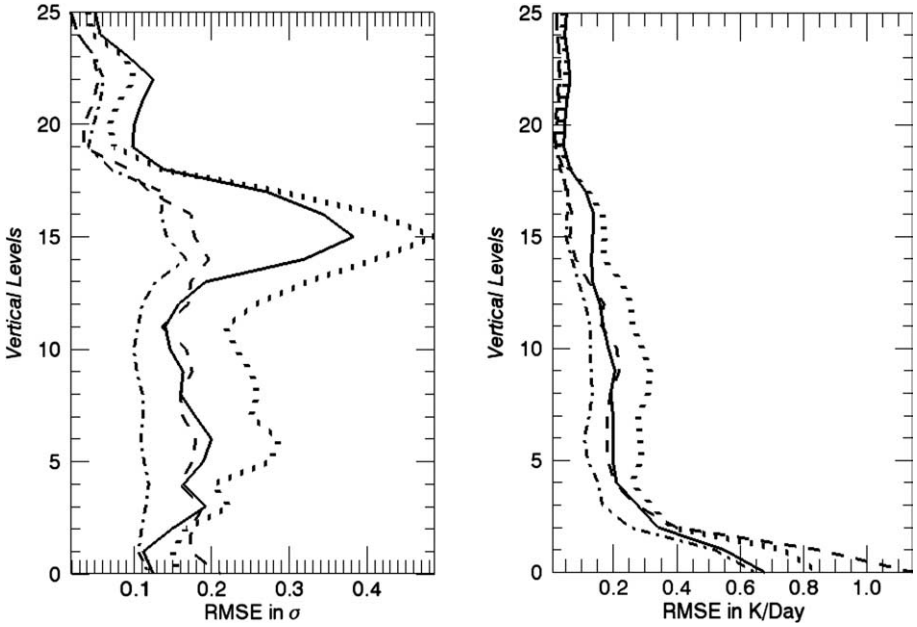

Fig. 5. LWR NN emulation errors for NCAR CAM. Vertical profiles of RMSEs (5) of LWR HRs at each of 26 model vertical levels for different architectures of LWR NN emulation and for different types of output normalizations. The left panel shows relative RMSEs in units of the standard deviations of LWR HRs calculated at the same vertical level. The right panel shows the absolute RMSEs in K/day. Dash-dotted lines show RMSEs for a battery of 33 single-output NNs. Solid, dashed, and dotted lines show RMSEs for three single NNs with 33 outputs and with 150 hidden neurons each. Dotted lines correspond to the $[-1,1]$ outputs normalization, dashed and solid curves correspond to the normalizations (7) and (8), respectively.

additional flexibility can be effectively used for various applications.

4.2. Normalization of NN outputs

For a NN (2) with a single linear output, normalization of the output is not a complicated problem. Any traditional normalization, like normalizing to the interval [–1,1] or using the following normalization

$$y' = \frac{y-\bar{y}}{\sigma} \tag{6}$$

where $\bar{y}$ is the mean value of y and σ is its standard deviation, leads to similar approximation errors.

For a single NN with multiple outputs, the normalization of outputs affects the approximation accuracy more significantly. Fig. 5 illustrates the dependence of approximation errors, at the different vertical model levels, on the type of the output normalization. The right panel shows the absolute approximation RMSEs for LWR heating rates in K/day; the left panel shows the relative approximation RMSEs normalized at each vertical level using the standard deviation (σ_q) of the heating rate at this level, q. The dotted curve corresponds to the $[-1,1]$ output normalization. It is clear that this normalization deemphasizes the contribution of the vertical levels with small (levels 13–18) and large (levels 0–3) σ_q, in the error function leading to larger and vertically nonuniform absolute and relative errors. The normalization similar to (6) for the case of multiple outputs can be written as

$$y_q = \alpha \frac{y_q-\bar{y}_q}{\sigma_q} \tag{7}$$

where $\alpha \leq 1$ is introduced to accelerate the training of linear weights in the output layer of NN (2). In the case of multiple outputs, this normalization leads to very different approximation errors (dashed curves) as compared with the $[-1,1]$ normalization. The normalization (7) leads to a more uniform vertical distribution of relative errors (left panel) and significantly reduces relative and absolute errors at vertical levels with small σ_q (levels 13–18); however, it significantly increases the errors at vertical levels with large σ_q (levels 0–3). A similar distribution of errors is produced by the battery of 33 single-output NNs that uses normalization (6). In this case, the smaller errors are due to the significantly larger total number of hidden neurons.

A compromise between the $[-1,1]$ normalization and the normalization (7) can be reached using the following normalization,

$$y'_q = \alpha \frac{y_q-\bar{y}_q}{\sigma} \tag{8}$$

where σ is the standard deviation for all outputs (heating rates at all vertical levels). The errors for this normalization are also shown in Fig. 5 (solid curves). For different applications of the NN emulations, different types of error distribution may be desirable; smaller absolute or relative errors may be preferable. Different output normalizations in the case of a single emulating NN with multiple outputs may provide a tool for managing this kind of requirement.

4.3. Training algorithm

In HGCM applications, the dimensionality of the training space reaches 10^5–10^6. It is our experience that in this case

the most reliable and stable (but slow) algorithm for NN training is back-propagation with adjustable learning rates and reasonable criteria for automatic stopping. We use back-propagation for training both linear and nonlinear weights in NN (2). NN weights are adjusted after each training record; however, criteria for a learning rate adjustment and automatic stopping are calculated and checked after each training epoch. Criteria used for changing learning rates include the magnitudes of: (1) a relative error, (2) an increment of an relative error, and (3) relative changes in NN weights. Predefined values of the same characteristics serve as automatic stopping criteria; in addition, the training stops if an error does not improve after a predefined number of training epochs.

4.4. Initialization of NN weights

It is well known that initialization of weights during the nonlinear optimization (the MLP training) can affect the resulting solution for NN weights (and hence generalization). A nonlinear error or lost function has multiple local minima. The back-propagation algorithm, as almost any algorithm available for solving a nonlinear optimization problem (NN training), converges to a local minimum. Usually, multiple initializations (even multiple initialization procedures) (e.g. Nguyen & Widrow, 1990; Wessels & Bernard, 1992) are applied allowing to avoid shallow local minima and to choose a local minimum with a sufficiently small error. In our applications, multi-collinearities (inter-correlations) in the input data and their high dimensionality partly alleviate the problem of seeking for a local minimum with a small error among multiple local minima. Multi-collinearities in the input data lead to equalization of local minima especially in the case of higher input dimensionality. From the point of view of the approximation problem, all these local minima give almost equally good solutions because the approximation errors for these minima are almost equally small. However, for our applications, we expect our NN emulations to provide a smooth interpolation in addition to good approximation. Thus, we apply different initializations that lead to different local minima, different solutions for the NN weights, and different interpolations. Then we test the quality of the interpolation using the test set and running different NN emulations in HGCM in parallel with the original GCM.

4.5. Generation of representative data sets for training

NN emulations in HGCMs are expected to be used to produce climate simulations, weather forecasts, etc. for periods from several days (for weather prediction) to several decades or hundreds of years (for climate simulations and predictions). During such long periods of integration, NN emulations can be applied up to about 10^{12} times and many new atmospheric states, that were not present in the dataset used during the training process, may occur in the simulations. This means that, during this long period of integration, NN emulation is used for massive interpolation and even for limited extrapolation. NN emulation is supposed to do this interpolation or limited extrapolation with high accuracy (taking into account our main requirement of providing a close similarity of the results generated by the HGCM to those of the original GCM). NN (MLP) is a very good tool for approximation; it is good enough for a limited generalization, including moderate interpolation and limited extrapolation. NN is not a very powerful tool for extended generalization, including far extrapolation and filling large gaps in the input space. These difficulties make the creation of a representative data set for NN training a non-trivial problem. However, in our case, many difficulties in this task are alleviated by using easily available simulated data for training. Moreover, because our training set consists of input/output profiles distributed more or less uniformly all over the globe, with all seasonal variations included, our data set is designed to be representative in terms of a complete set of physical states presented in data. It even includes some redundant data that makes it suitable for training a NN with good interpolation properties. In our applications, using data simulated during a 1 year GCM run for NN training allows us to produce a follow-up HGCM integration with this NN emulation for at least 20–40 years without losing simulation accuracy and integration stability. The accuracy of the HGCM simulation is high practically everywhere with only the potential of small error increases in some limited areas. If a training set is enriched by data from the second year of simulation, both the simulation accuracy in these areas and the error uniformity improve significantly. We experimented with the simulation length needed for creating a representative data set and concluded that a 2-year simulation is enough, especially for satisfying the specific conditions in these areas.

5. Discussions and future plans

Let us discuss some selected topics relevant to our applications that are still in the process of development and investigation.

5.1. Dynamical adjustment and quality control

The NN emulation approach described in this paper depends significantly on our ability to generate a representative training set that avoids using NN for extrapolation far beyond the domain covered by the training set and for interpolation into large gaps inside the domain. Taking into account that the input domain dimensionality is of the order of several hundred or more, it is rather difficult to cover the entire domain, especially its ‘far corners’ associated with rear events, even when we use the simulated data for NN training. Another related problem arises because NN emulates the part of a climate change simulated by GCM. It means that, in the process of running a climate simulation, the domain configuration may change as compared to its configuration when the training set was generated (e.g. climate change scenario). In both described situations the emulating NN may be forced to extrapolate beyond its generalization ability, which may lead to errors

in emulating NN outputs that may affect the corresponding HGCM simulations.

To take care of these kinds of problems and make our NN emulation approach suitable for long-term climate change simulations and for operational use in weather prediction, we are developing two new techniques: a *compound parameterization* (CP) and an NN *dynamical adjustment* (DA). Here, we will just briefly outline them.

The compound parameterization consists of three elements: the original parameterization, its NN emulation, and a quality control block (QCB). During a routine HGCM simulation with CP or HGCM/CP, the NN emulation is used by default and generates physical parameters (outputs) that are checked by QCB. If QCB accepts the parameters they are used in the HGCM. If QCB rejects the parameters generated by the NN emulation, the original parameterization is used instead for generating physical parameters to be used by the HGCM. At the same time, inputs and outputs of the original parameterization used as substitutions for the QCB-rejected parameters are saved for a follow-up adjustment of the NN emulation. After accumulating a sufficient number of records, a dynamical adjustment of the NN emulation is produced using a short retraining using the accumulated input/output records. Thus, the adopted NN emulation is dynamically adjusted to the changes and/or new events produced by the complex environmental model.

There are several possible designs to consider for the QCB. The first and simplest QCB design is based on a set of regular physical and statistical tests for the spatial, temporal, and internal consistency of the NN outputs. The second design is based on training additional NNs specifically for estimating the errors in the NN emulation outputs. If these errors exceed a predefined threshold, the original parameterization is used instead of NN emulation. The third and probably most promising design is based on the domain check technique proposed in the context of NN applications in satellite remote sensing (Krasnopolsky & Schiller, 2003). In this case, the QCB is a combination of forward and inverse NNs. We have already applied this approach as a preliminary study in the ocean wave model (Tolman & Krasnopolsky, 2004).

5.2. Calculating NN Jacobian

The parameterization Jacobian, a matrix of first derivatives of parameterization outputs over inputs, may be useful in many cases. For example, in data assimilation applications (an optimal blending of observational and simulated data to produce the best possible blended data) a Jacobian is used to create an adjoint (the tangent–linear approximation). A Jacobian is also instrumental for a statistical analysis of the original parameterization and its NN emulation (sensitivity, robustness, and error propagation analyses). An inexpensive computation of the Jacobian when using NN emulation is one of the advantages of the NN approach. Using this Jacobian in combination with the tangent–linear approximation can additionally accelerate calculations (Krasnopolsky et al., 2002). However, since the Jacobian is not trained it is simply calculated through direct differentiation of an emulating NN. Generally speaking, in this case the statistical inference of a Jacobian is an ill-posed problem and it is not guaranteed that the derivatives will be sufficiently accurate.

As was mentioned above, after training using a redundant training set, our NN emulations demonstrated very good interpolation properties during long-term decadal integrations of the HGCM. This implies that, on average, the derivatives of these emulations are sufficiently accurate to provide a satisfactory interpolation. However, for large NNs (Chevalier & Mahfouf, 2001), such an accuracy of an instantaneous NN Jacobian may not be sufficient for using the Jacobian in the tangent–linear approximation. For this type of an application, our NN emulation approach that treats a parameterization as a single object offers a simple and straightforward solution that alleviates the need for calculating the NN Jacobian explicitly. The adjoint tangent–linear approximation of a parameterization (e.g. of a radiation parameterization) may be considered as an independent/new parameterization and our NN emulation approach can be applied to such new parameterization.

For other applications that require an explicit calculation of the NN Jacobian, several solutions have been offered and investigated: (1) the mean Jacobian can be calculated and used (Chevalier & Mahfonf 2001); (2) the Jacobian can be included in the training data set and as actual additional outputs from the NN; (3) the Jacobian can be trained as a separate additional NN (Krasnopolsky et al., 2002) (generation of a data set for training a Jacobian or an adjoint NN is not a significant problem in our case because simulated data are used); (4) regularization techniques like ‘weight smoothing’ (Aires, Schmitt, Chedin, & Scott, 1999) or the technique based on a principle component decomposition (Aries, Prigent, & Rossow, 2004) can be used to stabilize the Jacobians; (5) the error (or cost) function, which is minimized in the process of NN training, can be modified to accommodate the Jacobian (Lee & Oh, 1997). In other words, the Euclidian norm, which is usually used for calculating the error function, should be changed to the first order Sobolev norm. With such a change, the NN is trained to approximate not only the parameterization (as with the Euclidian norm) but also the parameterization’s first derivatives. This solution does not change the number of the NN outputs; however, it may require using more hidden neurons and may significantly complicate the minimization during the training since the complexity of the error function increases. This solution also requires the availability of an extended training set that includes first derivatives. Finally, it should be mentioned that the Jacobian modeling for large NNs still remains an open issue.

5.3. Investigation of alternative machine learning techniques

In this paper, only one MLT, the NN technique, has been discussed and investigated. We started from NNs because this technique is well established and developed. However, it is not optimal; there are new techniques like support vector machines (SVM) and related approaches which may provide an optimal approximation for obtaining MLT components with even better

accuracy and performance. We plan on investigating other MLTs in the future.

6. Conclusions

In the study, we introduced a new practical application of NN techniques to complex environmental numerical modeling. Using NNs allowed us to design and formulate a new paradigm in environmental numerical modeling. We introduced a new type of a complex hybrid environmental numerical model—a HGCM—based on a synergetic combination of deterministic modeling and the machine learning techniques within such a model. This approach uses neural networks as a statistical or machine learning technique to develop highly accurate and fast emulations for the slowest deterministic model components (model physics parameterizations). Then these NN emulations are combined with the remaining deterministic components of the original GCM to constitute a HGCM, which produces simulations that are very close to those of the original GCM but significantly faster. The synergy of this approach leads to new opportunities in environmental modeling due to the use of HGCMs: (1) more frequent calculations of the model physics that improves the temporal consistency of the model physics and the model dynamics calculations; (2) a possibility of using new more sophisticated physical parameterizations that are currently computationally prohibited; (3) introducing models with higher resolutions; and (4) producing ensemble climate simulations and weather predictions for reduction of their uncertainties.

In some cases, the original deterministic component (parameterization) in a GCM is so complex that it is not practical to emulate it as a single mapping. In those cases, the idea of combining deterministic and NN components can be applied not to the entire GCM but to a single component (parameterization) of the model in order to develop a hybrid parameterization. Such an approach has already been successfully applied by Chevalier et al. (1998, 2000) and has been used since the fall of 2002 in the operational data assimilation system at the European Centre for Medium-range Weather Forecasts.

As we mentioned above, computations of physical processes are not the only computational 'bottlenecks' in GCMs. Including chemical, biological, hydrological and other processes into GCMs provides similar (sometimes even greater) computational challenges. The new paradigm of a hybrid model, combining deterministic and machine learning or statistical components, can provide fast and accurate calculations of these processes as well. The developed hybrid modeling paradigm and related NN emulation approach we developed are applicable, in our view, to other similar complex numerical models used outside the field of environmental modeling applications such as complex models in computational physics, chemistry, biology, etc.

Acknowledgements

The authors would like to thank Drs D. Chalikov and H. Tolman for fruitful cooperation, Drs F. Chevallier, W. Collins, P. Rasch, J. Tribbia, S. Lord, and D.B. Rao for useful discussions, and Mr A. Belochitski for his useful contributions to the study and programming support.

References

Aires, F., Prigent, C., & Rossow, W. B. (2004). Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3. Network Jacobians. *Journal of Geophysical Research*, *109*, D10305.

Aires, F., Schmitt, M., Chedin, A., & Scott, N. (1999). The 'weight smoothing' regularization of MLP for Jacobian stabilization. *IEEE Transactions on Neural Networks*, *10*, 1502–1510.

Chevallier, F., Chéruy, F., Scott, N. A., & Chédin, A. (1998). A neural network approach for a fast and accurate computation of longwave radiative budget. *Journal of Applied Meteorology*, *37*, 1385–1397.

Chevallier, F., & Mahfouf, J.-F. (2001). Evaluation of the Jacobians of infrared radiation models for variational data assimilation. *Journal of Applied Meteorology*, *40*, 1445–1461.

Chevallier, F., Morcrette, J.-J., Chéruy, F., & Scott, N. A. (2000). Use of a neural-network-based longwave radiative transfer scheme in the EMCWF atmospheric model. *Quarterly Journal of the Royal Meteorological Society*, *126*, 761–776.

Chou, M. -D., Suarez, M. J., Liang, X. -Z., & Yan, M. M. -H. (2001). A thermal infrared radiation parameterization for atmospheric studies. In: Max J. Suarez (Ed.), *Technical report series on global modeling and data assimilation, NASA/TM-2001-104606* (*Vol. 19*).

Collins, W. D. (2001). Parameterization of generalized cloud overlap for radiative calculations in general circulation models. *Journal of the Atmospheric Sciences*, *58*, 3224–3242.

Collins, W. D., Hackney, J. K., & Edwards, D. P. (2002). A new parameterization for infrared emission and absorption by water vapor in the National Center for Atmospheric Research Community Atmosphere Model. *Journal of Geophysical Research*, *107*(D22), 1–20.

Cybenko, G. (1989). Approximation by superposition of sigmoidal functions. *Mathematics of Control Signals and Systems*, *2*, 303–314.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*, 183–192.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward network. *Neural Networks*, *4*, 251–257.

Journal of Climate, 11(6) (the special issue) (1998).

Krasnopolsky, V., Breaker, L. C., & Gemmill, W. H. (1995). A neural network as a nonlinear transfer function model for retrieving surface wind speeds from the special sensor microwave imager. *Journal of Geophysical Research*, *100*, 11,033–11,045.

Krasnopolsky, V., Breaker, L. C., & Gemmill, W. H. (1997). A neural network forward model for direct assimilation of SSM/I brightness temperatures into atmospheric models. *Research activities in atmospheric and oceanic modeling*. CAS/JSC working group on numerical experimentation, report no. 25, WMO/TD-No. 792, pp. 1.29–1.30.

Krasnopolsky, V., Chalikov, D. V., & Rao, D. B. (2000). Application of neural networks for efficient calculation of sea water density or salinity from the UNESCO equation of state. *Proceedings of the second conference on artificial intelligence* (pp. 27–30). Long Beach, CA: AMS.

Krasnopolsky, V., Gemmill, W. H., & Breaker, L. C. (1999). A multi-parameter empirical ocean algorithm for SSM/I retrievals. *Canadian Journal of Remote Sensing*, *25*, 486–503.

Krasnopolsky, V. M., Chalikov, D. V., & Tolman, H. L. (2002). A neural network technique to improve computational efficiency of numerical oceanic models. *Ocean Modelling*, *4*, 363–383.

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Chalikov, D. V. (2005). New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of long wave radiation in a climate model. *Monthly Weather Review*, *133*, 1370–1383.

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Chou, Ming-Dah (2005). Robustness of the NN approach to emulating atmospheric long wave

radiation in complex climate models. *Proceedings of the international joint conference on neural networks, July 31–August 4, 2005, Montréal, Qué., Canada* (pp. 2661–2665).

Krasnopolsky, V. M., & Schiller, H. (2003). Some neural network applications in environmental sciences. Part I: Forward and inverse problems in satellite remote sensing. *Neural Networks*, *16*, 321–334.

Lee, J. W., & Oh, J.-H. (1997). Hybrid learning of mapping and its Jacobian in multilayer neural networks. *Neural Computation*, *9*, 937–958.

Nguyen, D., & Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *Proceedings of the international joint conference of neural networks* (Vol. 3) (pp. 21–26).

Schoendorf, J., Rabitz, H., & Li, G. (2003). A fast and accurate operational model of ionospheric electron density. *Geophysical Research Letters*, *30*, 1492–1495.

Tolman, H. L., & Krasnopolsky, V. M. (2004). Nonlinear interactions in practical wind wave models. *Eighth international workshop on wave hindcasting and forecasting, Turtle Bay, Hawaii, CD-ROM, E.1.*

Tolman, H. L., Krasnopolsky, V. M., & Chalikov, D. V. (2005). Neural network approximations for nonlinear interactions in wind wave spectra: Direct mapping for wind seas in deep water. *Ocean Modelling*, *8*, 253–278.

Wessels, L. F. A., & Bernard, E. (1992). Avoiding false local minima by proper initialization of connections. *IEEE Transactions on Neural Networks*, *3*, 899–905.